Remarks:

A. As to the objections to the drawings:

(1) Applicant states that the specification does make reference to element reference nos. 511 (see paragraph 92) and 512 (see paragraph 95).

(2) Applicant states that there is no conflict in the way that ref. nos. 908 and 909 are used (see Fig. 9). Ref. no. 909 refers to the entire packet (see paragraph 144), while ref. no. 908 refers to the data portion of the packet(see paragraph 149, referring to the size, that is, the number of data bytes, defined by ref. no. 908).

B. As to the claims:

The present invention is essentially a method for storing nested data structures which minimizes disk input and output. Nowhere in the prior art is there any indication of the operation which would support nested data structures and the storage of such structures. The main benefit of the present invention is the efficient creation, editing, and storage of nested data structures.

The prior art referred to in the Office Action does not mention or imply nested data structures. Each of the patents mentioned explicitly refer to relational databases (in the case of Gioielli, Gajda, and Jorgensen), and non-relational sources defined as keyed, sequential and binary file systems (in Gajda). None of these prior art data models support nested data

structures. In fact, relational databases are by definition non-nested (i.e. the table elements must only consist of atomic values). File systems are also not nested, in that they consist only of a keyed or sequential sequence of bytes (i.e. atomic values).

Claims 6 and 7 (and 11 and 12) add the idea of data nesting to the storage method claimed in claims 3-5. Accordingly, new claim 13 includes all the limitations of claims 3 through 5, and adds the limitations of claims 6 and 7 to show the creation of a nested data structure. This is accomplished by having component packet records which may contain an indexed/keyed collection of other component packets. Note that by being self-referential, these claims define an arbitrarily nested storage method/system.

Second, the storage method/system packs the data in such a way that when the size permits, the whole nested structure can be read in one disk access, and that as the number and size of the data increases, only those data elements that grow past a certain size and number will require additional disk accesses. This method relies on the addition of the logical block number to each component packet record.

Because the prior art databases and file systems do not allow nested data structures, there is no mention of efficient storage of such structures in database or file system publications nor in the patents referred to by the Office Action.

As an example of the difference between nested and non-

nested data structures, consider a database comprised of a table of employee records and a table of document references. Each employee record could be keyed by employee number and could be comprised of data values, like last name, first name, age, salary, perhaps a text field describing job function. Further suppose the document records are keyed by a unique document id and are comprised of a document name and a file system reference. Assume the file system contains the document contents.

The above is a description of a traditional non-nested data structure discussed in the prior art. Each record is comprised of atomic data values, i.e. is a collection of strings, numbers, external references, or a file of bytes. To create a nested data structure we need to extend the employee record by adding a data value representing a non-atomic value, such as an employee's contact information. Contact information is in general a collection of phone numbers, email addresses, physical addresses, etc.

In traditional databases, such a collection would require the definition of an additional table of records, with one record for each piece of contact information. Reading an employee's contact information would require separate access to this table for retrieval, since it is stored independently of the employee record. A list of an employee's documents would similarly require a separate table and separate access to acquire the list.

In a nested implementation, the employee record contains the

contact information directly and could be read with the same disk access that retrieves the employee record itself. This is one of the benefits of nested data structures. A document list could also be stored directly inside the employee record.

A nested data model and storage method is depicted in figure 9 of the patent application, and is explained in the detailed description as described *infra*. Figure 9 shows a top-level component packet 901 and component data 903, which is the entry point to a keyed search structure which itself contains other component packets 909. This nested component packet may contain, among other things, the entry point into another keyed search structure, which may contain, among other things, more component packets. The claims allow this nesting to go on to an arbitrary number of levels.

The purpose of the logical block number is covered in paragraphs 104 to 126 of the detailed description. Adding a logical block number to the component packet record allows record data to be stored in an alternate block. This allows changes to the size and number of a component packet's records without an equivalent change in the size of the component packet itself.

This is important in the storage of nested structures, since without such a mechanism, any change in the size of an inner data value would otherwise ripple upward/outward in the nesting hierarchy and force an equivalent change in the size of the enclosing component packet. Allowing component data to move to

another block breaks this propagation by allowing the parent component to remain the roughly the same size in spite of changes to inner data records. This reduces the cost of adding or deleting new data into a nested data structure and is a key method for providing efficient storage of nested data.

The new independent claims 13 and 19 combine the limitations of claims 3 through 7, plus adding additional elements which provide for the creation, editing, and storing of nested data structures. The additional dependent claims use new language, and make it clear that the ability to access the entire disk of the present data structure is dependent upon the unique architecture of the database of the present invention.

In summary, we agree with the Examiner that the claims are obvious when taken individually, but when taken together they describe nested data structures and associated storage methods not covered in the prior art. In order to make this fact clearer, the claims have been rewritten as described above.

In light of the amendment submitted above, and the arguments offered, it is believe that the present application is now in condition for allowance, which is hereby requested.

Respectfully submitted on February 20, 2007 by Attorney for the Applicant.

Mark P. White

Mark P. White

Reg. No. 37,757
White & Fudala, LC
57 Bedford St., Suite 100
Lexington, MA 02420
781-863-2041

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this correspondence (Amendment) is being facsimile transmitted to the United States Patent and Trademark Office, to FAX NUMBER: 571-273-8300.

on February 20, 2007

Mark P. White

Mark P. White
Reg. No. 37,757